

**ANL252 (Online)**

**PYTHON FOR DATA ANALYTICS**

# **End-of-Course Assessment**

**January 2023 Presentation**

**Submitted by:**

|  |  |
| --- | --- |
| **Name** | **PI No.** |
| **Brandan Chia Jian Hao** | **Y2172066** |
| Click to enter NAME. | Click to enter PI NO. |
| Click to enter NAME. | Click to enter PI NO. |
| Click to enter NAME. | Click to enter PI NO. |

**Tutorial Group: ­­­­­­­­­­­­ T 05**

**Instructor’s Name: Dr. Munish Kumar**

**Submission Date: 06/03/2023**

**Question 1**

Variables that contain missing values are Claim\_ID, Actual & Terms.

**Question 2**

We should replace all missing data with 0 or ignore the missing data. “Claim\_ID” is a unique identifier and it make no sense in replacing it with mean, median, mode and replacing values from previous or next ones. “Actual” could be before or after the planned date of claim settlement and it make no sense in filling the missing values with mean, median or mode. “Terms” depending on individuals and what is the actual terms and condition during the purchase of policy affecting the claim amount could not be filled up based on mean, median or mode and replacing values from previous or next ones. We should not remove data because there might be claimant and auditors requesting for past data.

**Question 3**

To change the datatype of “Planned”, “Actual” & “Created” into datetime64[ns] and “Amount” to float as we might need to calculate difference in date and interquartile range.

“Amount” should be in numeric therefore find all non-numeric values and replace it with correct number.

Outliers are data that, in addition to missing data, may lead to bias in the estimation of statistical parameters and, ultimately, the goodness of fit of the models. Before starting any study, it is crucial to recognize biased estimates and make the necessary corrections because they should be avoided.

**Question 4**

Insurance company process claims fast.

Chart, scatter chart

Description automatically generated

The Category, Terms and type affecting the claim amount.

The Region whereby claims submitted affecting the claim amount.

**Question 5**

The independent variable x will be the amount, category, terms, region and type while the dependent variable y being the delay in days.

**Question 6**

Y = mX + b

References:

What is Claims Processing? Definition & How it Works - Attune Technologies Pvt Ltd. (2018, October 16). Attune Technologies Pvt Ltd. http://attunelive.com/claims-processing-definition-works/

Wu, K. Y. (2022). ANL252 Python for data analytics (study guide). Singapore University of Social Sciences.

# **Appendix**

# import libraries

import pandas as pd

import numpy as np

import seaborn as sns

import matplotlib.pyplot as plt

# Making a list of missing value types

missing\_values = ["Unkn", "???",""]

# read excel file located on my desktop

df = pd.read\_csv("C:\\Users\\shado\\OneDrive\\Desktop\\ECA.csv", na\_values = missing\_values)

df.shape

df.isnull()

#calculate the sum of missing values in each columns

df.isnull().sum()

#prints information about the DataFrame

df.info()

#replace missing data with 0

df.fillna(0)

#change column type

convert\_dict = {'Claim\_ID': object,

'Policy\_No': object,

}

df = df.astype(convert\_dict)

df['Planned'] = pd.to\_datetime(df['Planned'])

df['Actual'] = pd.to\_datetime(df['Actual'])

df['Created'] = pd.to\_datetime(df['Created'])

print(df.dtypes)

#finding non numeric value in Amount

print (df[pd.to\_numeric(df['Amount'], errors='coerce').isnull()])

#replace value

df['Amount'] = df['Amount'].replace(['1762.OO'], ['1762.00'])

#change column type

df = df.astype({'Amount':'float'})

print(df.dtypes)

#IQR method to find outlier

q1 = df["Amount"].quantile(0.25)

q3 = df["Amount"].quantile(0.75)

iqr = q3-q1

q1,q3,iqr

upper\_limit = q3 + (1.5\* iqr)

lower\_limit = q1 - (1.5\* iqr)

lower\_limit, upper\_limit

#find outliers

df.loc[(df["Amount"] > upper\_limit) | (df["Amount"] < lower\_limit)]

#delete outlier data

new\_df = df.loc[(df["Amount"] < upper\_limit) & (df["Amount"] > lower\_limit)]

print("before removing outliers:",len(df))

print("after removing outliers:",len(new\_df))

print("outliers:",len(df)-len(new\_df))

#create new columns that contains date differences

new\_df['diff\_days'] = (new\_df['Actual'] - new\_df['Planned']) / np.timedelta64(1, 'D')

new\_df

new\_df.plot(kind='scatter', x='diff\_days', y='Amount', figsize=(9,6), color='Green')

#select columns with index positions

returns = new\_df.iloc[:, [6, 12]]

returns.isnull().sum()

returns\_records\_dropped = returns.dropna(axis=0, how='any')

returns\_records\_dropped.info()

#Linear regression

X = returns.iloc[:, :-1].values

y = returns.iloc[:, 1].values

from sklearn.preprocessing import LabelEncoder, OneHotEncoder

label\_X = LabelEncoder()

X[:,0] = label\_X.fit\_transform(X[:,0])

encoder = OneHotEncoder(categorical\_features = [0])

X = encoder.fit\_transform(X).toarray()

from sklearn.cross\_validation import train\_test\_split

X\_train, X\_test, y\_train,y\_test = train\_test\_split(X, y, test\_size = 0.4, random\_state = 0)

#fitting Simple Regression to training set

from sklearn.linear\_model import LinearRegression

regressor = LinearRegression()

regressor.fit(X\_train, y\_train)

#predecting the test set results

y\_pred = regressor.predict(X\_test)